# IT 140 Design Document Template

## Instructions

Fill out the sections below. Be sure to remove the bracketed text before submitting your Design Document.

Angel Davila

## Storyboard (Description and Map)

The story is that you are are a demon slayer trying to find the king of the demons to defeat it. There are demons in each wing, so 7 in total, and each one drops a health item which increases your overall health. Once you’ve defeated all 7, the king of demons shows up in the dungeon, and you have to defeat it to win.

East

North

West

East

South

West

East

South

North

West

East

North

South

West

Central Wing

West Wing

South Wing

North Wing

South East Wing

North East Wing

East Wing

Dungeon

## I think before we create the logic that handles moving between rooms, we might want to store the demons, king of demons and the player in some sort of class, this way we can store the game state. In this case, we might want to store, players health, positions, items. Maybe something along these lines of:

Class Player:

Name string

Health int

Position int

Class Demon:

Health int

Position int

Item int

Class King:

Health int

Position int

We could get a little more complex with the game and store some sort of attack or attack damage that the villain can inflict on the player, but in this case, we are going to just keep all character health’s in the class. To not over complicate the positioning of characters, we can store this data in an enum and create a function to handle the change of a position.

Class Rooms(Enum):

CENTRAL\_WING = 1

NORTH\_WING = 2

WEST\_WING = 3

SOUTH\_WING = 4

EAST\_WING = 5

SOUTH\_EAST\_WING = 6

NORTH\_EAST\_WING = 7

DUNGEON = 8

When the game starts, we might want to create all 7 demon instances and store them in rooms 1 to 7.

## Pseudocode or Flowchart for Code to “Move Between Rooms”

The movement should be its own function which does all the handling of room movements.

Function player\_movement(player\_object, movement\_option):

GET player\_position from player\_object

CHECK if the movement\_option is valid from the current position

LONG IF statement handling the possibilities to allow to move

Movement options could be another enum, something like:

Class Movement(Enum):

UP = 1

DOWN = 2

LEFT = 3

RIGHT = 4

## Pseudocode or Flowchart for Code to “Get an Item”

This will be handled once the player has defeated the demon. But we can have this inside of a function that handles a fight with a villain.

Function initiate\_fight(player, villan):

This function will handle all the fighting

After the fight outcome, there needs to be a way to drop the health item that will be added to the player

Function drop\_item(player):

This will handle the drop item, essentially giving the player a random set of health

ADD player health

Function main():

Will contain the main loop of the function.

Take in player options and call functions that need to be called.